Operations on Automata with All States Final

Article *in* Electronic Proceedings in Theoretical Computer Science · May 2014

DOI: 10.4204/EPTCS.151.14 · Source: arXiv

Galina Jiraskova
Slovak Academy of Sciences
96 PUBLICATIONS **1,197** CITATIONS
SEE PROFILE

Peter Mlynárčik
University of Presov in Presov
12 PUBLICATIONS **58** CITATIONS
SEE PROFILE

Matúš Palmovský
Slovak Academy of Sciences
6 PUBLICATIONS **35** CITATIONS
SEE PROFILE

Operations on Automata with All States Final

Kristína Čevorová
Mathematical Institute,
Slovak Academy of Sciences,
Bratislava, Slovakia
cevorova@mat.savba.sk

Galina Jirásková*
jiraskov@saske.sk

Peter Mlynárčik
Mathematical Institute,
Slovak Academy of Sciences,
Košice, Slovakia
mlynarcik1972@gmail.com

Matúš Palmovský
matp93@gmail.com

Juraj Šebej†
Institute of Computer Science,
Šafárik University,
Košice, Slovakia
juraj.sebej@gmail.com

We study the complexity of basic regular operations on languages represented by incomplete deterministic or nondeterministic automata, in which all states are final. Such languages are known to be prefix-closed. We get tight bounds on both incomplete and nondeterministic state complexity of complement, intersection, union, concatenation, star, and reversal on prefix-closed languages.

1 Introduction

A language L is prefix-closed if $w \in L$ implies that every prefix of w is in L. It is known that a regular language is prefix-closed if and only if it is accepted by a nondeterministic finite automaton (NFA) with all states final [18]. In the minimal incomplete deterministic finite automaton (DFA) for a prefix-closed language, all the states are final as well.

The authors of [18] examined several questions concerning NFAs with all states final. They proved that the inequivalence problem for NFAs with all states final is PSPACE-complete in the binary case, but polynomially solvable in the unary case. Next, they showed that minimizing a binary NFA with all states final is PSPACE-hard, and that deciding whether a given NFA accepts a language that is not prefix-closed is PSPACE-complete, while the same problem for DFAs can be solved in polynomial time. The NFA-to-DFA conversion and complementation of NFAs with all states final have been also considered in [18], and the tight bound 2^n for the first problem, and the lower bound 2^{n-1} for the second one have been obtained.

The quotient complexity of prefix-closed languages has been studied in [5]. The quotient of a language L by the string w is the set $L_w = \{x \mid wx \in L\}$. The quotient complexity of a language L, $\kappa(L)$, is the number of distinct quotients of L. Quotient complexity is defined for any language, and it is finite if and only if the language is regular. The quotient automaton of a regular language L is the DFA $(\{L_w \mid w \in \Sigma^*\}, \Sigma, \cdot, L_\varepsilon, F)$, where $L_w \cdot a = L_{wa}$, and a quotient L_w is final if it contains the empty string. The quotient automaton of L is a minimal complete DFA for L, so quotient complexity is the same as the state complexity of L which is defined as the number of states in the minimal DFA for L. In [5], the tight bounds on the quotient complexity of basic regular operation have been obtained, and to prove upper bounds, the properties of quotients have been used rather than automata constructions.

[*] Research supported by grant APVV-0035-10.
[†] Research supported by grant VEGA 1/0479/12.

Z. Ésik and Z. Fülöp (Eds.): Automata and Formal Languages 2014 (AFL 2014)
EPTCS 151, 2014, pp. 201–215, doi:10.4204/EPTCS.151.14

Automata with all states final represent systems, for example, production lines, and their intersection or parallel composition represents the composition of these systems [21]. A question that arises here is, whether the complexity of intersection of automata with all states final is the same as in the general case of arbitrary DFAs or NFAs. At the first glance, it seems that this complexity could be smaller. Our first result shows that this is not the case. We show that both incomplete and nondeterministic state complexity of intersection on prefix-closed languages is given by the function mn, which is the same as in the general case of regular languages.

In the deterministic case, to have all the states final, we have to consider incomplete deterministic automata because otherwise, the complete automaton with all states final would accept the language consisting of all the strings over an input alphabet. Notice that the model of incomplete deterministic automata has been considered already by Maslov [20]. The same model has been used in the study of the complexity of the shuffle operation [6]; here, the complexity on complete DFAs is not known yet.

We next study the complexity of complement, union, concatenation, square, star, and reversal on languages represented by incomplete DFAs or NFAs with all states final. We get tight bounds in both nondeterministic and incomplete deterministic cases. In the nondeterministic case, all the bounds are the same as in the general case of regular languages, except for the bound for star that is n instead of $n+1$. However, to prove the tightness of these bounds, we usually use larger alphabets than in the general case of regular languages where all the upper bounds can be met by binary languages [10, 12].

To get lower bounds, we use a fooling-set lower-bound method [1, 2, 3, 8, 11]. In the case of union and reversal, the method does not work since it provides a lower bound on the size of NFAs with multiple initial states. Since the nondeterministic state complexity of a regular language is defined using a model of NFAs with a single initial state [10], we have to use a modified fooling-set technique to get the tight bounds $m+n+1$ and $n+1$ for union and reversal, respectively.

In the case of incomplete deterministic finite automata, the tight bounds for complement, union, concatenation, star, and reversal are $n+1, mn+m+n, m\cdot 2^{n-1}+2^n-1$, 2^{n-1}, and 2^n-1, respectively. To define worst-case examples, we use a binary alphabet for union, star, and reversal, and a ternary alphabet for concatenation.

The paper is organized as follows. In the next section, we give some basic definitions and preliminary results. In Sections 3 and 4, we study boolean operations. Concatenation is discussed in Section 5, and star and reversal in Section 6. The last section contains some concluding remarks.

2 Preliminaries

In this section, we recall some basic definitions and preliminary results. For details and all unexplained notions, the reader may refer to [24].

A *nondeterministic finite automaton* (NFA) is a quintuple $A=(Q,\Sigma,\delta,I,F)$, where Q is a finite set of states, Σ is a finite alphabet, $\delta\colon Q\times\Sigma\to 2^Q$ is the transition function which is extended to the domain $2^Q\times\Sigma^*$ in the natural way, $I\subseteq Q$ is the set of initial states, and $F\subseteq Q$ is the set of final states. The language accepted by A is the set $L(A)=\{w\in\Sigma^*\mid \delta(I,w)\cap F\neq\emptyset\}$.

The *nondeterministic state complexity* of a regular language L, $\operatorname{nsc}(L)$, is the smallest number of states in any NFA with a *single initial state* recognizing L.

An NFA A is *incomplete deterministic* (DFA) if $|I|=1$ and $|\delta(q,a)|\leq 1$ for each q in Q and each a in Σ. In such a case, we write $\delta(q,a)=q'$ instead of $\delta(q,a)=\{q'\}$. A non-final state q of a DFA is called a *dead* state if $\delta(q,a)=q$ for each symbol a in Σ.

The *incomplete state complexity* of a regular language L, $\text{isc}(L)$, is the smallest number of states in any incomplete DFA recognizing L. An incomplete DFA is minimal (with respect to the number of states) if it does not have any dead state, all its states are reachable, and no two distinct states are equivalent.

Every NFA $A = (Q, \Sigma, \delta, I, F)$ can be converted to an equivalent DFA $A' = (2^Q, \Sigma, \cdot, I, F')$, where $R \cdot a = \delta(R, a)$ and $F' = \{R \in 2^Q \mid R \cap F \neq \emptyset\}$. The DFA A' is called the *subset automaton* of the NFA A. The subset automaton need not be minimal since some of its states may be unreachable or equivalent. However, if for each state q of an NFA A, there exists a string w_q that is accepted by A only from the state q, then the subset automaton of the NFA A does not have equivalent states since if two subsets of the subset automaton differ in a state q, then they are distinguishable by w_q.

To prove the minimality of NFAs, we use a fooling set lower-bound technique, see [1, 2, 3, 8, 11].

Definition A set of pairs of strings $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ is called a *fooling set* for a language L if for all i, j in $\{1, 2, \ldots, n\}$, the following two conditions hold:

(F1) $x_i y_i \in L$, and

(F2) if $i \neq j$, then $x_i y_j \notin L$ or $x_j y_i \notin L$.

It is well known that the size of a fooling set for a regular language provides a lower bound on the number of states in any NFA (with multiple initial states) for the language. The argument is simple. Fix the accepting computations of any NFA on strings $x_i y_i$ and $x_j y_j$. Then, the states on these computations reached after reading x_i and x_j must be distinct, otherwise the NFA accepts both $x_i y_j$ and $x_j y_i$ for two distinct pairs. Hence we get the following observation.

Lemma 1 ([3, 8, 11])**.** *Let $\mathscr{F}$ be a fooling set for a language L. Then every NFA (with multiple initial states) for the language L has at least $|\mathscr{F}|$ states.* □

The next lemma shows that sometimes, if we insist on having a single initial state in an NFA, one more state is necessary. It can be used in the case of union, reversal, cyclic shift [15], and AFA-to-NFA conversion [13]. In each of these cases, NFAs with a single initial state require one more state than NFAs with multiple initial states. For the sake of completeness, we recall the proof of the lemma here.

Lemma 2 ([14])**.** *Let $\mathscr{A}$ and $\mathscr{B}$ be sets of pairs of strings and let u and v be two strings such that $\mathscr{A} \cup \mathscr{B}$, $\mathscr{A} \cup \{(\varepsilon, u)\}$, and $\mathscr{B} \cup \{(\varepsilon, v)\}$ are fooling sets for a language L. Then every NFA with a single initial state for the language L has at least $|\mathscr{A}| + |\mathscr{B}| + 1$ states.*

Proof. Consider an NFA for a language L, and let $\mathscr{A} = \{(x_i, y_i) \mid i = 1, 2, \ldots, m\}$ and $\mathscr{B} = \{(x_{m+j}, y_{m+j}) \mid j = 1, 2, \ldots, n\}$. Since the strings $x_k y_k$ are in L, we fix an accepting computation of the NFA on each string $x_k y_k$. Let p_k be the state on this computation that is reached after reading x_k. Since $\mathscr{A} \cup \mathscr{B}$ is a fooling set for L, the states $p_1, p_2, \ldots, p_{m+n}$ are pairwise distinct. Since $\mathscr{A} \cup \{(\varepsilon, u)\}$ is a fooling set, the initial state is distinct from all the states $p_1, p_2, \ldots, p_m$. Since $\mathscr{B} \cup \{(\varepsilon, v)\}$ is a fooling set, the (single) initial state is also distinct from all the states $p_{m+1}, p_{m+2}, \ldots, p_{m+n}$. Thus the NFA has at least $m + n + 1$ states. □

Example Let $K = (a^3)^*$ and $L = (b^3)^*$. Then $\text{nsc}(K) = 3$ and $\text{nsc}(L) = 3$, and the language $K \cup L$ is accepted by a 6-state NFA with two initial states. Therefore, we cannot expect that we will be able to find a fooling set for $K \cup L$ of size 7. However, every NFA with a *single* initial state for the language $K \cup L$ requires at least 7 states since Lemma 2 is satisfied for the language $K \cup L$ with

$$\begin{aligned}
\mathscr{A} &= \{(a, a^2), (a^2, a), (a^3, a^3)\},\\
\mathscr{B} &= \{(b, b^2), (b^2, b), (b^3, b^3)\},\\
u &= b^3, \text{ and}\\
v &= a^3.
\end{aligned}$$

If $w = uv$ for strings u and v, then u is a *prefix* of w. A language L is *prefix-closed* if $w \in L$ implies that every prefix of w is in L. The following observations are easy to prove.

Proposition 3 ([18])**.** *A regular language is prefix-closed if and only if it is accepted by some NFA with all states final.* □

Proposition 4. *Let A be a minimal incomplete DFA for a language L. Then the language L is prefix-closed if and only if all the states of the DFA A are final.* □

3 Complementation

If L is a language over an alphabet Σ, then the complement of L is the language $L^c = \Sigma^* \setminus L$. If L is accepted by a minimal complete DFA A, then we can get a minimal DFA for L^c from the DFA A by interchanging the final and non-final states. In the case of incomplete DFAs, we first have to add a dead state, that is, a non-final state which goes to itself on each input, and let all the undefined transitions go to the dead state. After that, we can interchange the final and non-final states to get a (complete) DFA for the complement. This gives the following result.

Theorem 5. *Let $n \geq 1$. Let L be a prefix-closed regular language over an alphabet Σ with $\text{isc}(L) = n$. Then $\text{isc}(L^c) \leq n+1$, and the bound is tight if $|\Sigma| \geq 1$.*

Proof. For tightness, we can consider the unary prefix-closed language $\{a^i \mid 0 \leq i \leq n-1\}$. □

If a language L is represented by an n-state NFA, then we first construct the corresponding subset automaton, and then interchange the final and non-final states to get a DFA for the language L^c of at most 2^n states. This upper bound on the nondeterministic state complexity of complement on regular languages is know to be tight in the binary case [12].

For prefix-closed languages, we get the same bound, however, to prove tightness, we use a ternary alphabet. Whether or not the bound 2^n can be met by a binary language remains open.

Theorem 6. *Let $n \geq 2$. Let L be a prefix-closed regular language over an alphabet Σ with $\text{nsc}(L) = n$. Then $\text{nsc}(L^c) \leq 2^n$, and the bound is tight if $|\Sigma| \geq 3$.*

Proof. The upper bound is the same as in the general case of regular languages [10]. To prove tightness, consider the language L accepted by the NFA N shown in Figure 1, in which state n goes to the empty set on both a and b, and to $\{1\}$ on c. Each other state i goes to $\{i+1\}$ on both a and c, and to $\{1, i+1\}$ on b. Our aim is to describe a fooling set $\mathscr{F} = \{(x_S, y_S) \mid S \subseteq \{1, 2, \ldots, n\}\}$ of size 2^n for L^c.

First, let us show that each subset of $\{1, 2, \ldots, n\}$ is reachable in the subset automaton of the NFA N. The initial state is $\{1\}$, and each singleton set $\{i\}$ is reached from $\{1\}$ by a^{i-1}. The empty set is reached from $\{n\}$ by a. The set $\{i_1, i_2, \ldots, i_k\}$ of size k, where $2 \leq k \leq n$ and $1 \leq i_1 < i_2 < \cdots < i_k \leq n$, is reached from the set $\{i_2 - i_1, \ldots, i_k - i_1\}$ of size $k-1$ by the string ba^{i_1-1}. This proves reachability by induction. Now, define x_S as the string, by which the initial state 1 of the NFA N goes to the set S.

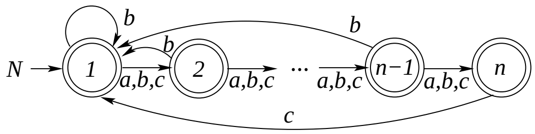

Figure 1: The NFA N of a prefix-closed language L with $\text{nsc}(L^c) = 2^n$.

Next, for a subset S of $\{1,2,\ldots,n\}$, define the string y_S as the string $y_S = y_0 y_1 \cdots y_{n-1}$ of length n, where

$$y_i = \begin{cases} a, & \text{if } n-i \in S, \\ c, & \text{if } n-i \notin S. \end{cases}$$

We claim that the string y_S is rejected by the NFA N from each state in S and accepted from each state that is not in S. Indeed, if i is a state in S, then $y_{n-i} = a$ and $y_S = uav$ with $u = y_0 y_1 \cdots y_{n-i-1}$ and $v = y_{n-i+1} y_{n-i+2} \cdots y_{n-1}$. Hence $|u| = n-i$, which means that the state i goes to $\{n\}$ by u since both a and c move each state q to state $q+1$. However, in state n the NFA N cannot read a, and therefore the string $y_S = uav$ is rejected from i. On the other hand, if $i \notin S$, then $y_{n-i} = c$, and the string $y_S = ucv$ with $|u| = n-i$ and $|v| = i-1$ is accepted from i through the computation $i \xrightarrow{u} n \xrightarrow{c} 1 \xrightarrow{v} i$.

Now, we are ready to prove that the set of pairs of strings $\mathscr{F} = \{(x_S, y_S) \mid S \subseteq \{1,2,\ldots,n\}\}$ is a fooling set for the language L^c.

(F1) By x_S, the initial state 1 goes to the set S. The string y_S is rejected by N from each state in S. It follows that the NFA N rejects the string $x_S y_S$. Thus the string $x_S y_S$ is in L^c.

(F2) Let $S \neq T$. Then without loss of generality, there is a state i such that $i \in S$ and $i \notin T$. By x_S, the initial state 1 goes to S, so it also goes to the state i. Since $i \notin T$, the string x_T is accepted by N from i. Therefore, the NFA N accepts the string $x_S y_T$, and so this string is not in L^c.

Hence $\mathscr{F}$ is a fooling set for L^c of size 2^n. By Lemma 1, we have $\operatorname{nsc}(L^c) \geq 2^n$. □

4 Intersection and Union

In this section, we study the incomplete and nondeterministic state complexity of intersection and union of prefix-closed languages. If regular languages K and L are accepted by m-state and n-state NFAs, respectively, then the language $K \cap L$ is accepted by an NFA of at most mn states, and this bound is known to be tight in the binary case [10]. Our first result shows that the bound mn can be met by binary prefix-closed languages. Then, using this result, we get the same bound on the incomplete state complexity of intersection on prefix-closed languages.

Theorem 7. *Let K and L be prefix-closed languages over an alphabet Σ with $\operatorname{nsc}(K) = m$ and $\operatorname{nsc}(L) = n$. Then $\operatorname{nsc}(K \cap L) \leq mn$, and the bound is tight if $|\Sigma| \geq 2$.*

Proof. The upper bound is the same as for regular languages [10]. For tightness, consider prefix-closed binary languages $K = \{w \in \{a,b\}^* \mid \#_a(w) \leq m-1\}$ and $L = \{w \in \{a,b\}^* \mid \#_b(w) \leq n-1\}$ that are accepted by an m-state and an n-state incomplete DFAs A and B, respectively, shown in Figure 2.

Consider the set of pairs of strings $\mathscr{F} = \{(a^i b^j, a^{m-1-i} b^{n-1-j}) \mid 0 \leq i \leq m-1, 0 \leq j \leq n-1\}$ of size mn. Let us show that $\mathscr{F}$ is a fooling set for the language $K \cap L$.

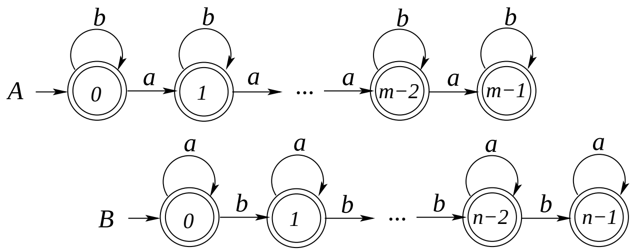

Figure 2: The incomplete DFAs A and B of prefix-closed languages K and L with $\operatorname{nsc}(K \cap L) = mn$.

(F1) The string $a^ib^j \cdot a^{m-1-i}b^{n-1-j}$ has exactly $m-1$ a's and $n-1$ b's. It follows that it is in $K \cap L$.

(F2) Let $(i,j) \neq (k,\ell)$. If $i<k$, then the string $a^kb^\ell \cdot a^{m-1-i}b^{n-1-j}$ contains $m-1+(k-i)$ a's, and therefore it is not in K. The case of $j<\ell$ is symmetric.

Hence $\mathscr{F}$ is a fooling set for $K \cap L$, and the theorem follows. □

Theorem 8. *Let K and L be prefix-closed languages over an alphabet Σ with $\operatorname{isc}(K)=m$ and $\operatorname{isc}(L)=n$. Then $\operatorname{isc}(K\cap L)\leq mn$, and the bound is tight if $|\Sigma|\geq 2$.*

Proof. Let $A=(Q_A,\Sigma,\delta_A,s_A,Q_A)$ and $B=(Q_B,\Sigma,\delta_B,s_B,Q_B)$ be incomplete DFAs for K and L, respectively. Define an incomplete product automaton $M=(Q_A\times Q_B,\Sigma,\delta,(s_A,s_B),Q_A\times Q_B)$, where

$$\delta((p,q),a)=\begin{cases}(\delta_A(p,a),\delta_B(q,a)), & \text{if both } \delta_A(p,a) \text{ and } \delta_B(q,a) \text{ are defined,}\\ \text{undefined}, & \text{otherwise}.\end{cases}$$

The DFA M accepts the language $K\cap L$. This gives the upper bound mn. For tightness, consider the same languages K and L as in the proof of the previous theorem. Notice that K and L are accepted by m-state and n-state incomplete DFAs, respectively. We have shown that nondeterministic state complexity of their intersection is mn. It follows that the incomplete state complexity is also at least mn. □

Our next result on the incomplete state complexity of union on prefix-closed languages can be derived from the result on the quotient complexity of union in [5]. For the sake of completeness, we restate it in terms of incomplete complexities, and recall the proof.

Theorem 9. *Let K and L be prefix-closed languages over an alphabet Σ with $\operatorname{isc}(K)=m$ and $\operatorname{isc}(L)=n$. Then $\operatorname{isc}(K\cup L)\leq mn+m+n$, and the bound is tight if $|\Sigma|\geq 2$.*

Proof. Let $A=(\{0,1,\dots,m-1\},\Sigma,\delta_A,0,F_A)$ and $B=(\{0,1,\dots,n-1\},\Sigma,\delta_B,0,F_B)$ be incomplete DFAs for the languages K and L, respectively. To construct a DFA for the language $K\cup L$, we first add the dead states m and n to the DFAs A and B, and let go all the undefined transitions to the dead states. Now we construct the classic product-automaton from the resulting complete DFAs with the state set $\{0,1,\dots,m\}\times\{0,1,\dots,n\}$. All its states are final, except for the state (m,n) that is dead, and we do not count it. Hence we get the upper bound $mn+m+n$ on the incomplete state complexity of union.

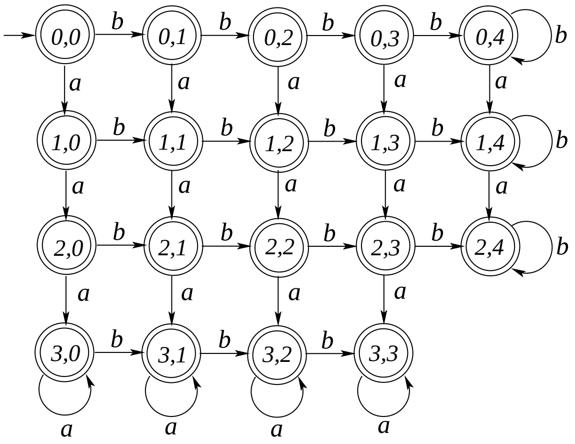

Figure 3: The product automaton for incomplete DFAs A and B from Figure 2; $m=3$ and $n=4$.

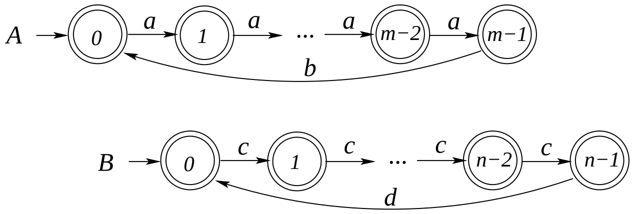

Figure 4: The NFAs A and B of prefix-closed languages K and L with $\operatorname{nsc}(K \cup L) = m+n+1$.

For tightness, we again consider the languages described in the proof of Theorem 7. We add the dead states m and n and construct the product automaton. The product automaton in the case of $m = 3$ and $n = 4$ is shown in Figure 3.

Each state (i, j) of the product automaton is reached from the initial state $(0,0)$ by the string $a^i b^j$. Let (i, j) and (k,ℓ) be two distinct states of the product automaton. If $i < k$, then the string $a^{m-k}b^n$ is rejected from (k,ℓ) and accepted from (i, j). If $j < \ell$, then the string $b^{n-\ell}a^m$ is rejected from (k,ℓ) and accepted from (i, j). Thus all the states in the product-automaton are reachable and pairwise distinguishable, and the lower bound $mn+m+n$ follows. □

In the nondeterministic case, the upper bound for union on regular language is $m+n+1$, and it is tight in the binary case [10]. We get the same bound for union on prefix-closed languages, however, to define witness languages, we use a four-letter alphabet.

Theorem 10. *Let K and L be prefix-closed languages over an alphabet Σ with $\operatorname{nsc}(K) = m$ and $\operatorname{nsc}(L) = n$. Then $\operatorname{nsc}(K \cup L) \le m+n+1$, and the bound is tight if $|\Sigma| \ge 4$.*

Proof. The upper bound is the same as for regular languages [10]. To prove tightness, let K and L be the prefix-closed languages accepted by the NFAs A and B, respectively, shown in Figure 4. Let

$$\mathscr{A} = \{(a^i, a^{m-1-i}b) \mid i = 1,2,\dots,m-1\} \cup \{(a^{m-1}b, a)\},$$
$$\mathscr{B} = \{(c^j, c^{n-1-j}d) \mid j = 1,2,\dots,n-1\} \cup \{(c^{n-1}d, c)\}.$$

Let us show that $\mathscr{A} \cup \mathscr{B}$ is a fooling set for the language $K \cup L$.

(F1) We have $a^i \cdot a^{m-1-i}b = a^{m-1}b$ and $c^j \cdot c^{n-1-j}d = c^{n-1}d$. Both these strings are in $K \cup L$. The strings $a^{m-1}b \cdot a$ and $c^{n-1}d \cdot c$ are in $K \cup L$ as well.

(F2) If $1 \le i < i' \le m-1$, then the string $a^i \cdot a^{m-1-i'}b$ is not in K since $m-1-(i'-i) < m-1$. Next, if $1 \le i \le m-1$, then $a^{m-1}b \cdot a^{m-1-i}b$ is not in K. The argumentation for two pairs from $\mathscr{B}$ is similar. If we concatenate the first part of a pair in $\mathscr{A}$ with the second part of a pair in $\mathscr{B}$, then we get a string that either contains all three symbols a, c, d, or contains both symbols a and d. No such string is in $K \cup L$.

Thus $\mathscr{A} \cup \mathscr{B}$ is a fooling set for the language $K \cup L$. Moreover, the sets $\mathscr{A} \cup \{(\varepsilon, c)\}$ and $\mathscr{B} \cup \{(\varepsilon, a)\}$ are fooling sets for $K \cup L$ as well. By Lemma 2, we have $\operatorname{nsc}(K \cup L) \ge m+n+1$. □

5 Concatenation

In this section, we deal with the concatenation operation on prefix-closed languages. We start with incomplete state complexity. We use a slightly different ternary witness language than in [5], and prove the upper bound using automata constructions.

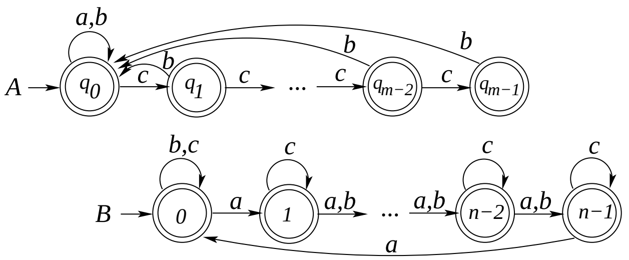

Figure 5: The incomplete DFAs A and B of languages K and L with $\operatorname{isc}(KL) = m \cdot 2^{n-1} + 2^n - 1$.

Theorem 11. *Let $m, n \geq 3$. Let K and L be prefix-closed languages over an alphabet Σ with $\operatorname{isc}(K) = m$ and $\operatorname{isc}(L) = n$. Then $\operatorname{isc}(KL) \leq m \cdot 2^{n-1} + 2^n - 1$, and the bound is tight if $|\Sigma| \geq 3$.*

Proof. Let $A = (Q_A, \Sigma, \delta_A, s_A, Q_A)$ and $B = (Q_B, \Sigma, \delta_B, s_B, Q_B)$ be incomplete DFAs with all states final accepting the languages K and L, respectively. Construct an NFA N for the language KL from the DFAs A and B by adding the transition on a symbol a from a state q in Q_A to the initial state s_B of B whenever the transition on a in state q is defined in A. The initial states of the NFA N are s_A and s_B, and the set of final states is Q_B. Each reachable subset of the subset automaton of the NFA N contains at most one state of Q_A, and several states of Q_B. Moreover, if a state of Q_A is in a reachable subset S, then S must contain the state s_B. This gives the upper bound $m \cdot 2^{n-1} + 2^n - 1$ on $\operatorname{isc}(KL)$ since the empty set is not counted.

For tightness, consider the prefix-closed languages K and L accepted by incomplete DFAs A and B, respectively, shown in Figure 5, in which the transitions are as follows:

on a, state q_0 goes to itself, and each state j goes to $(j+1) \bmod n$;

on b, each state q_i goes to state q_0, state 0 goes to itself, and state j with $1 \leq j \leq n-2$ goes to $j+1$;

on c, each state q_i with $0 \leq i \leq m-2$ goes to q_{i+1}, and each state j goes to itself;

and all the remaining transitions are undefined.

Construct an NFA N for the language KL as described above. Let us show that the subset automaton of the NFA N has $m \cdot 2^{n-1} + 2^n - 1$ reachable and pairwise distinguishable non-empty subsets.

(1) First, let us show that each set $\{q_0\} \cup S$ is reachable, where $S \subseteq \{0, 1, \ldots, n-1\}$ and $0 \in S$. The proof is by induction on the size of subsets. The set $\{q_0, 0\}$ is the initial subset. The set $\{q_0, 0, j_1, j_2, \ldots, j_k\}$ with $1 \leq j_1 < j_2 < \cdots < j_k \leq n-1$ is reached from the set $\{q_0, 0, j_2 - j_1, \ldots, j_k - j_1\}$ by the string $ab^{j_1 - 1}$, and the latter set is reachable by induction.

(2) Now, let us show that each set $\{q_i\} \cup S$, is reachable, where $1 \leq i \leq m-1$, $S \subseteq \{0, 1, \ldots, n-1\}$ and $0 \in S$. The set $\{q_i\} \cup S$ is reached from $\{q_0\} \cup S$ by c^i, and the latter set is reachable as shown in (1).

(3) Next, we show that each set S with $S \subseteq \{0, 1, \ldots, n-1\}$ and $0 \in S$ is reachable. The set S is reached from $\{q_{m-1}\} \cup S$ by c, and the latter set is reachable as shown in case (2).

(4) Finally, we show that each non-empty set S with $S \subseteq \{0, 1, \ldots, n-1\}$ and $0 \notin S$ is reachable. If $S = \{j_1, j_2, \ldots, j_k\}$ with $j_1 \geq 1$, then S is reached from the set $\{0, j_2 - j_1, \ldots, j_k - j_1\}$ by a^{j_1}, and the latter set is reachable as shown in case (3).

This proves the reachability of $m \cdot 2^{n-1} + 2^n - 1$ non-empty subsets.

To prove distinguishability, notice that the string b^n is accepted by the DFA B only from the state 0, and the string $a^{n-1-i}ab^n$ is accepted only from the state i ($1 \leq i \leq n-1$). If S and T are two distinct subsets of $\{0, 1, \ldots, n-1\}$, then S and T differ in a state i. If $i = 0$, then b^n distinguishes S and T, and if $i \geq 1$, then $a^{n-i}b^n$ distinguishes S and T.

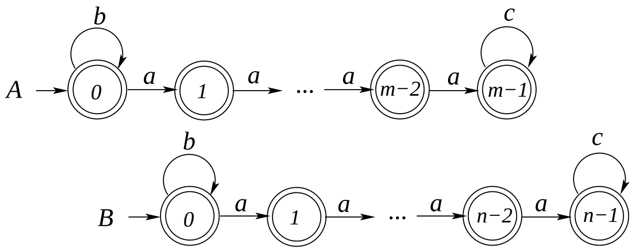

Figure 6: The incomplete DFAs of prefix-closed languages K and L with $\operatorname{nsc}(KL) = m+n$.

Next, the sets $\{q_i\} \cup S$ and $\{q_i\} \cup T$, where S and T are distinct subsets of $\{0,1,\dots,n-1\}$, go to S and T, respectively, by c^m. Since S and T are distinguishable, the sets $\{q_i\} \cup S$ and $\{q_i\} \cup T$ are distinguishable as well.

Finally, notice that the string $b^n ab^n$ is accepted by the NFA N from each state q_i, but rejected from each state i in $\{0,1,\dots,n-1\}$. Hence the sets $\{q_i\} \cup S$ and T, where S and T are subsets of $\{0,\dots,n-1\}$, are distinguishable. Now let $0 \le i < j \le m-1$. Then $\{q_i\} \cup S$ and $\{q_j\} \cup T$ go to $\{q_{i+m-j}\} \cup S$ and T, respectively, by c^{m-j}. Since $\{q_{i+m-j}\} \cup S$ and T are distinguishable, the sets $\{q_i\} \cup S$ and $\{q_j\} \cup T$ are distinguishable as well. This proves the distinguishability of all the reachable subsets, and completes the proof. □

In the next theorem, we consider the nondeterministic case. For regular languages, the upper bound on the nondeterministic state complexity of concatenation is $m+n$, and it is tight in the binary case [10]. For prefix-closed languages, we get the same bound for concatenation. However, we define witness languages over a ternary alphabet.

Theorem 12. *Let $m,n \ge 3$. Let K and L be prefix-closed languages over an alphabet Σ with $\operatorname{nsc}(K) = m$ and $\operatorname{nsc}(L) = n$. Then $\operatorname{nsc}(KL) \le m+n$, and the bound is tight if $|\Sigma| \ge 3$.*

Proof. The upper bound is the same as for regular languages [10]. For tightness, consider the ternary prefix-closed languages K and L accepted by incomplete DFAs A and B, respectively, shown in Figure 6. Notice that if a string w is in KL, then w is in the language $b^*a^*c^*b^*a^*c^*$, and the number of a's in w is at most $(n+m-2)$.

For $i = 0,1,\dots,m+n-1$, define the pair (x_i, y_i) as follows:

$$(x_i,y_i) = (a^i, a^{m-1-i}cba^{n-1}), \quad \text{for } i = 0,1,\dots m-1,$$
$$(x_{m+j},y_{m+j}) = (a^{m-1}cba^j, a^{n-1-j}), \quad \text{for } j = 0,1,\dots n-1.$$

Let us show that the set of pairs $\mathscr{F} = \{(x_i,y_i) \mid i = 0,1,\dots,m+n-1\}$ is a fooling set for the language KL.

(F1) For each i, we have $x_iy_i = a^{m-1}cba^{n-1}$. Thus x_iy_i is in KL since $a^{m-1}c$ is in K and ba^{n-1} is in L.

(F2) Let $i < j$ and $(i,j) \ne (m-1,m)$. Then the number of a's in the string x_jy_i is greater than $m+n-2$, and therefore the string x_jy_i is not in KL. If $(i,j) = (m-1,m)$, then $x_my_{m-1} = a^{m-1}cbcba^{n-1}$. Thus x_my_{m-1} is not in $b^*a^*c^*b^*a^*c^*$, and therefore it is not in KL.

Hence the set $\mathscr{F}$ is a fooling set for the language KL, so $\operatorname{nsc}(KL) \ge m+n$. □

6 Star and Reversal

We conclude our paper with the star and reversal operation on prefix-closed languages. The star of a language L is the language $L^* = \bigcup_{i \ge 0} L^i$, where $L^0 = \{\varepsilon\}$ and $L^{i+1} = L^i \cdot L$.

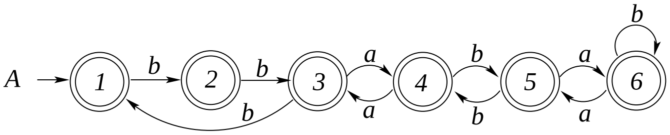

Figure 7: The incomplete DFA A of a prefix-closed language L with $\text{isc}(L^*) = 2^{n-1}$; $n = 6$.

If a regular language L is accepted by a complete n-state DFA, then the language L^* is accepted by a DFA of at most $3/4 \cdot 2^n$ states, and the bound is tight in the binary case [20, 25].

For prefix-closed languages, the upper bound on the quotient complexity for star is $2^{n-2}+1$, and it has been shown to be tight in the ternary case [5]. In the case of incomplete state complexity, we get the bound 2^{n-1}. For the sake of completeness, we give a simple proof of the upper bound using automata constructions. Moreover, we are able to define a witness language over a binary alphabet.

Theorem 13. *Let $n \geq 4$. Let L be a prefix-closed regular language over an alphabet Σ with $\text{isc}(L) = n$. Then $\text{isc}(L^*) \leq 2^{n-1}$, and the bound is tight if $|\Sigma| \geq 2$.*

Proof. Let $A = (Q, \Sigma, \cdot, s, Q)$ be an incomplete DFA for L. Construct an NFA A^* for L^* from the DFA A by adding the transition on a symbol a from a state q to the initial state s whenever the transition $q \cdot a$ is defined. In the subset automaton of the NFA A^*, each reachable set is either empty, or it contains the initial state s. It follows that $\text{isc}(L^*) \leq 2^{n-1}$.

For tightness, consider the binary incomplete DFA with the state set $\{1, 2, \ldots, n\}$, the initial state 1 and with all states final. The transitions are as follows. By a, the transitions in states 1 and 2 are undefined, each odd state i with $3 \leq i \leq n-1$ goes to $i+1$, and each even state i with $3 \leq i \leq n-1$ goes to $i-1$. By b, there is a cycle $(1,2,3)$, each odd state i with $4 \leq i \leq n-1$ goes to $i-1$, and each even state i with $4 \leq i \leq n-1$ goes to $i+1$. If n is odd, then n goes to itself by a, otherwise it goes to itself by b. The DFA for $n = 6$ is shown in Figure 7.

Notice that each state i with $3 \leq i \leq n$ has exactly one in-transition on a and on b. Denote by $a^{-1}(i)$ the state that goes to i on a, and by $b^{-1}(i)$ the state that goes to i on b.

Construct an NFA A^* as described above. Let us show that in the subset automaton of the NFA A^*, all subsets of $\{1, 2, \ldots, n\}$ containing state 1 are reachable and pairwise distinguishable.

We prove reachability by induction on the size of subsets. The basis is $|S| = 1$, and the set $\{1\}$ is reachable since it is the initial state of the subset automaton. Assume that every set S containing 1 with $|S| = k$, where $1 \leqslant k \leqslant n-1$, is reachable. Let $S = \{1, i_1, i_2, i_3, \ldots, i_k\}$, where $2 \leqslant i_1 < i_2 < \cdots < i_k \leqslant n$, be a set of size $k+1$. Consider three cases:

(i) $i_1 = 2$. Take $S^{'} = \{1, b^{-1}(i_2), b^{-1}(i_3), \ldots, b^{-1}(i_k)\}$. Then $|S^{'}| = k$, and therefore $S^{'}$ is reachable by the induction hypothesis. Since we have $S' \xrightarrow{b} \{1, 2, i_2, \ldots, i_k\} = S$, the set S is reachable.

(ii) $i_1 = 3$. Take $S' = \{1, 2, b^{-1}(i_2), b^{-1}(i_3), \ldots, b^{-1}(i_k)\}$. Then $|S'| = k+1$ and S' contains states 1 and 2. Therefore, the set S' is reachable as shown in case (i). Since we have $S^{'} \xrightarrow{b} \{1, 2, 3, i_2, i_3, \ldots, i_k\} \xrightarrow{aa} \{1, 3, i_2, i_3, \ldots, i_k\} = S$, the set S is reachable.

(iii) Let $i_1 = j \geq 3$, and assume that each set $\{1, j, i_2, \ldots, i_k\}$ is reachable. Let us show that then also each set $\{1, j+1, i_2, \ldots, i_k\}$ is reachable. If j is odd, then the set $\{1, j+1, i_2, \ldots, i_k\}$ is reached from the set $\{1, j, a^{-1}(i_2), a^{-1}(i_3), \ldots, a^{-1}(i_k)\}$ by a. If j is even, then the set $\{1, j+1, i_2, \ldots, i_k\}$ is reached from the set $\{1, j, b^{-1}(i_2), b^{-1}(i_3), \ldots, b^{-1}(i_k)\}$ by baa.

This proves reachability. To prove distinguishability, notice that the string $(ab)^{n-2}$ is accepted by the NFA A^* from state 3 since state 3 goes to the initial state 1 by $(ab)^{n-2}$ through the computation

$$3 \xrightarrow{ab} 5 \xrightarrow{ab} 7 \xrightarrow{ab} \cdots \xrightarrow{ab} n \xrightarrow{a} n \xrightarrow{b} n-1 \xrightarrow{ab} n-3 \xrightarrow{ab} \cdots \xrightarrow{ab} 4 \xrightarrow{ab} 1$$

if n is odd, and through a similar computation if n is even. On the other hand, the string $(ab)^{n-2}$ cannot be read from any other state $2i$ with $2 \le i \le n/2$ since we have

$$2i \xrightarrow{ab} \{2i-2,1,2\} \xrightarrow{ab} \{2i-4,1,2\} \xrightarrow{ab} \cdots \xrightarrow{ab} \{4,1,2\} \xrightarrow{a} \{3,1\} \xrightarrow{b} \{1,2\} \xrightarrow{ab} \emptyset,$$

thus $2i$ goes to the empty set by $(ab)^i$, so also by $(ab)^{n-2}$. If n is odd, then we have

$$2i+1 \xrightarrow{ab} \{2i+3,1,2\} \xrightarrow{ab} \{2i+5,1,2\} \xrightarrow{ab} \cdots \xrightarrow{ab} \{n,1,2\} \xrightarrow{a} \{n,1\} \xrightarrow{b} \{n-1,1,2\} \xrightarrow{ab}$$

$$\{n-3,1,2\} \xrightarrow{ab} \cdots \xrightarrow{ab} \{2i,1,2\} \xrightarrow{(ab)^i} \emptyset,$$

thus $2i+1$ goes to the empty set by $(ab)^{n-i}$, $i \ge 2$, and so also by $(ab)^{n-2}$. For n even, the argument is similar. The string $(ab)^{n-2}$ is not accepted from states 1 and 2. Hence the NFA A^* accepts the string $(ab)^{n-2}$ only from the state 3. Since there is exactly one in-transition on b in state 3, and it goes from state 2, the string $b(ab)^{n-2}$ is accepted by A^* only from state 2. Similarly, the string $bb(ab)^{n-2}$ is accepted by A^* only from state 1. Next, for similar reasons, the string $a(ab)^{n-2}$ is accepted only from 4, the string $ba(ab)^{n-2}$ is accepted only from 5, and in the general case, the string $(ab)^i a(ab)^{n-2}$ is accepted only from $4+2i$ ($i \ge 0$), and the string $(ba)^i(ab)^{n-2}$ is accepted only from $3+2i$ ($i \ge 1$). Hence for each state q of the NFA A^*, there exists a string w_q that is accepted by A^* only from the state q. It follows that all the subsets of the subset automaton of the NFA A^* are pairwise distinguishable since two distinct subsets differ in a state q, and the string w_q distinguishes the two subsets. This completes the proof. □

We did some computations in the binary case. Having the files of n-state minimal binary pairwise non-isomorphic complete DFAs with a dead state and all the remaining states final, we computed the state complexity of the star of languages accepted by DFAs on the lists; here the state complexity of a regular language L, $\mathrm{sc}(L)$, is defined as the smallest number of states in any *complete* DFA for the language L. We computed the frequencies of the resulting complexities, and the average complexity of star. Our results are summarized in Table 2. Notice that for $n = 3,4,5$, there is just one language with $\mathrm{sc}(L) = n$ and $\mathrm{sc}(L^*) = 2$. Let us show that this holds for every n with $n \ge 3$.

$n \backslash \mathrm{sc}(L^*)$	1	2	3	4	5	6	7	8	9	average
2	-	2	-	-	-	-	-	-	-	2
3	8	1	6	-	-	-	-	-	-	1.866
4	161	1	48	30	6	-	-	-	-	1.857
5	4177	1	771	275	350	84	84	-	26	1.849

Table 1: The frequencies of the complexities and the average complexity of star on prefix-closed languages in the binary case; $n = 2,3,4,5$.

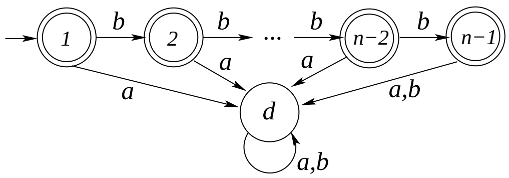

Figure 8: The only binary n-state complete DFA of a prefix-closed language L with $\mathrm{sc}(L^*) = 2$.

Proposition 14. *Let $n \geq 3$. There exists exactly one (up to renaming of alphabet symbols) binary prefix-closed regular language L with $\mathrm{sc}(L) = n$ and $\mathrm{sc}(L^*) = 2$.*

Proof. Let $A = (\{0,1\}, \{a,b\}, \delta, 0, F)$ be a minimal two-state DFA for the language L^*. Since L is prefix-closed, the language L^* is prefix-closed as well. It follows that state 0 is final, and state 1 is dead, thus $F = \{0\}$ and $\delta(1,a) = \delta(1,b) = 1$.

Without loss of generality, state 1 is reached from the initial state 0 by a, thus $\delta(0,a) = 1$.

Since $n \geq 3$, the language L contains a non-empty string. This means that the language L^* contains a non-empty string as well. Therefore, we must have $\delta(0,b) = 0$, and so $L^* = b^*$.

Now let B be the minimal n-state DFA for L. Then all the states of B are final, except for the dead state. Since $L^* = b^*$, no a may occur in any string of L. Hence each non-dead state of B must go to the dead state on a. Since all states must be reachable, we must have a path labeled by b^{n-2} and going through all the final states. The last final state must go to the dead state on b because otherwise all final states would be equivalent. The resulting n-state DFA B is shown in Figure 8. □

The reverse w^R of a string w is defined by $\varepsilon^R = \varepsilon$, and $(wa)^R = aw^R$ for a in Σ and w in Σ^*. The reverse of a language L is the language $L^R = \{w^R \mid w \in L\}$. If a regular language L is accepted by a complete n-state DFA, then the language L^R is accepted by a complete DFA of at most 2^n states [22, 25], and the bound is tight in the binary case [16, 19].

For prefix-closed languages, the quotient complexity of reversal is 2^{n-1} [5], and it follows from the results on ideal languages [4] since reversal commutes with complementation, and the complement of a prefix-closed language is a right ideal; here a language L is a right ideal if $L = L \cdot \Sigma^*$.

We restate the result for reversal in terms of incomplete state complexity, and prove tightness using a slightly different witness language.

Theorem 15. *Let $n \geq 2$. Let L be a prefix-closed regular language over an alphabet Σ with $\mathrm{isc}(L) = n$. Then $\mathrm{isc}(L^R) \leq 2^n - 1$, and the bound is tight if $|\Sigma| \geq 2$.*

Proof. Let A be an incomplete DFA for L. Construct an NFA A^R for the language L^R from the DFA A by swapping the role of the initial and final states, and by reversing all the transitions. The subset automaton of the NFA A^R has at most $2^n - 1$ non-empty reachable states, and the upper bound follows.

For tightness, consider the incomplete DFA A with all states final, shown in Figure 9. Construct an NFA A^R as described above. In the subset automaton of the NFA A^R, the initial state is $\{1,2,\ldots,n\}$. If S is a subset and if $i \in S$, then the subset $S \setminus \{i\}$ is reached from S by a^iba^{n-i}. This proves the reachability of all non-empty subset by odd induction. Since the states of the subset automaton of any reversed DFA are pairwise distinguishable [7, 16, 22], the theorem follows.

□

Now, let us turn to the nondeterministic case. For regular languages, the tight bound for both star and reversal is $n+1$. It is met by a unary language for star [10], and by a binary language for reversal [12].

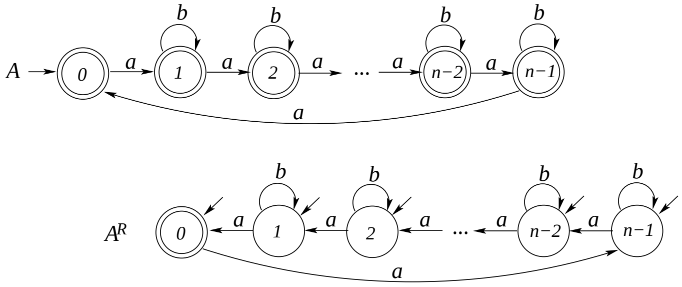

Figure 9: The incomplete DFA A of a language L with $\mathrm{isc}(L^R) = 2^n - 1$, and the NFA A^R.

For prefix-closed languages, we get the same bound for reversal. However, for star, the upper bound is n since every prefix-closed language contains the empty string, and there is no need to add a new initial state in the construction of an NFA for star. In the following theorem, we show that both these bounds are tight in the binary case.

Theorem 16. *Let $n \geq 2$. Let L be a prefix-closed language over an alphabet Σ with $\mathrm{nsc}(L) = n$. Then*
(1) $\mathrm{nsc}(L^*) \leq n$*,*
(2) $\mathrm{nsc}(L^R) \leq n+1$*,*
and both bounds are tight if $|\Sigma| \geq 2$.

Proof. (1) Let $N = (Q, \Sigma, \delta, s, F)$ be an n-state NFA for L. Since L is prefix-closed, the empty string is in L. Therefore, we can get an n-state NFA for the language L^* from the NFA N as follows: for each state q and each symbol a such that $\delta(q,a) \cap F \neq \emptyset$, we add a transition on a from q to the initial state s. Thus the upper bound is n.

For tightness, consider the prefix-closed language L accepted by the NFA A shown in Figure 10. Consider the set of pair of strings $\mathscr{F} = \{(a^i, a^{n-1-i}b) \mid i = 0,1,\ldots,n-1\}$ of size n. Let us show that $\mathscr{F}$ is a fooling set for the language L^*.

(F1) We have $a^i a^{n-1-i}b = a^{n-1}b$. Since the string $a^{n-1}b$ is in L, it also is in L^*.

(F2) Let $i < j$. Then $a^i a^{n-1-j}b = a^{n-1-(j-i)}b$. Since no string $a^\ell b$ with $\ell < n-1$ is in L, the string $a^{n-1-(j-i)}b$ is not in L^*.

Hence the set $\mathscr{F}$ is a fooling set for the language L^*, and the lower bound follows.

(2) The upper bound is the same as for regular languages [10]. It is shown in [12, Theorem 2] that this bound is met by the binary prefix-closed language L accepted by the NFA shown in Figure 10. The proof in [12] is by a counting argument. Notice that Lemma 2 is satisfied for the language L^R with $\mathscr{A} = \{(ba^i, a^{n-1-i}) \mid i = 0,1,\ldots,n-2\}, \mathscr{B} = \{(ba^{n-1}, ba^{n-1})\}, u = ba^{n-1}$, and $v = a$. This gives $\mathrm{nsc}(L^R) \geq n+1$ immediately. □

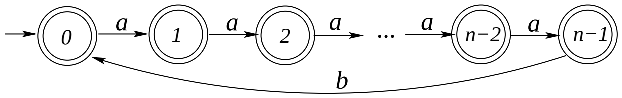

Figure 10: The NFA of a prefix-closed language L with $\mathrm{nsc}(L^*) = n$ and $\mathrm{nsc}(L^R) = n+1$.

		complement	$\lvert\Sigma\rvert$	intersection	$\lvert\Sigma\rvert$	union	$\lvert\Sigma\rvert$
isc	on prefix-closed	$n+1$	1	mn	2	$mn+m+n$	2
sc	on prefix-closed [5]	n	1	$mn-m-n+2$	2	mn	2
sc	on regular [20, 25]	n	1	mn	2	mn	2
nsc	on prefix-closed	2^n	3	mn	2	$m+n+1$	4
nsc	on regular [10, 12]	2^n	2	mn	2	$m+n+1$	2

Table 2: The complexity of boolean operations on prefix-closed and regular languages.

		concatenation	$\lvert\Sigma\rvert$	star	$\lvert\Sigma\rvert$	reversal	$\lvert\Sigma\rvert$
isc	on prefix-closed	$m2^{n-1}+2^n-1$	3	2^{n-1}	2	2^n-1	2
sc	on prefix-closed [5]	$(m+1)2^{n-2}$	3	$2^{n-2}+1$	3	2^{n-1}	2
sc	on regular [20, 25]	$m2^n-2^{n-1}$	2	$2^{n-1}+2^{n-2}$	2	2^n	2
nsc	on prefix-closed	$m+n$	3	n	2	$n+1$	2
nsc	on regular [10, 12]	$m+n$	2	$n+1$	2	$n+1$	2

Table 3: The complexity of concatenation, star, and reversal on prefix-closed and regular languages.

7 Conclusions

In this paper we considered operations on languages recognized by incomplete deterministic or nondeterministic finite automata with all states final. Our results are summarized in Tables 2 and 3. The results on quotient (state) complexity on prefix-closed languages are from [5], and the results for regular languages are from [10, 12, 20, 25]. Notice that in the nondeterministic case, our results are the same as in the general case of regular languages, except for the star operation. However, to prove tightness, we usually used larger alphabets than in the general case. Whether or not these bounds are tight also for smaller alphabets remains open.

References

[1] Alfred V. Aho, Jeffrey D. Ullman & Mihalis Yannakakis (1983): *On notions of information transfer in VLSI circuits*. In Johnson et al. [17], pp. 133–139. Available at `http://doi.acm.org/10.1145/800061.808742`.

[2] Jean-Camille Birget (1992): *Intersection and union of regular languages and state complexity*. *Inform. Process. Lett.* 43(4), pp. 185–190. Available at `http://dx.doi.org/10.1016/0020-0190(92)90198-5`.

[3] Jean-Camille Birget (1993): *Partial orders on words, minimal elements of regular languages and state complexity*. *Theoret. Comput. Sci.* 119(2), pp. 267–291. Available at `http://dx.doi.org/10.1016/0304-3975(93)90160-U`.

[4] Janusz A. Brzozowski, Galina Jirásková & Baiyu Li (2013): *Quotient complexity of ideal languages*. *Theoret. Comput. Sci.* 470, pp. 36–52. Available at `http://dx.doi.org/10.1016/j.tcs.2012.10.055`.

[5] Janusz A. Brzozowski, Galina Jirásková & Chenglong Zou (2014): *Quotient complexity of closed languages*. *Theory Comput. Syst.* 54(2), pp. 277–292. Available at `http://dx.doi.org/10.1007/s00224-013-9515-7`.

[6] Cezar Câmpeanu, Kai Salomaa & Sheng Yu (2002): *Tight lower bound for the state complexity of shuffle of regular languages*. *J. Autom. Lang. Comb.* 7(3), pp. 303–310.

[7] J.-M. Champarnaud, A. Khorsi & T. Paranthoën: *Split and join for minimizing: Brzozowski's algorithm*. Available at `http://jmc.feydakins.org/ps/c09psc02.ps`.

[8] Ian Glaister & Jeffrey Shallit (1996): *A lower bound technique for the size of nondeterministic finite automata*. *Inform. Process. Lett.* 59(2), pp. 75–77. Available at `http://dx.doi.org/10.1016/0020-0190(96)00095-6`.

[9] Edward A. Hirsch, Juhani Karhumäki, Arto Lepistö & Michail Prilutskii, editors (2012): *Computer Science - Theory and Applications - 7th International Computer Science Symposium in Russia, CSR 2012, Nizhny Novgorod, Russia, July 3-7, 2012. Proceedings*. *Lecture Notes in Computer Science* 7353, Springer. Available at `http://dx.doi.org/10.1007/978-3-642-30642-6`.

[10] Markus Holzer & Martin Kutrib (2003): *Nondeterministic descriptional complexity of regular languages*. *Internat. J. Found. Comput. Sci.* 14(6), pp. 1087–1102. Available at `http://dx.doi.org/10.1142/S0129054103002199`.

[11] Juraj Hromkovic (1997): *Communication complexity and parallel computing*. Springer. Available at `http://dx.doi.org/10.1007/978-3-662-03442-2`.

[12] Galina Jirásková (2005): *State complexity of some operations on binary regular languages*. *Theoret. Comput. Sci.* 330(2), pp. 287–298. Available at `http://dx.doi.org/10.1016/j.tcs.2004.04.011`.

[13] Galina Jirásková (2012): *Descriptional complexity of operations on alternating and boolean automata*. In Hirsch et al. [9], pp. 196–204. Available at `http://dx.doi.org/10.1007/978-3-642-30642-6_19`.

[14] Galina Jirásková & Tomáš Masopust (2011): *Complexity in union-free regular languages*. *Internat. J. Found. Comput. Sci.* 22(7), pp. 1639–1653. Available at `http://dx.doi.org/10.1142/S0129054111008933`.

[15] Galina Jirásková & Alexander Okhotin (2008): *State complexity of cyclic shift*. *RAIRO Theor. Inform. Appl.* 42(2), pp. 335–360. Available at `http://dx.doi.org/10.1051/ita:2007038`.

[16] Galina Jirásková & Juraj Šebej (2012): *Reversal of binary regular languages*. *Theoret. Comput. Sci.* 449, pp. 85–92. Available at `http://dx.doi.org/10.1016/j.tcs.2012.05.008`.

[17] David S. Johnson, Ronald Fagin, Michael L. Fredman, David Harel, Richard M. Karp, Nancy A. Lynch, Christos H. Papadimitriou, Ronald L. Rivest, Walter L. Ruzzo & Joel I. Seiferas, editors (1983): *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, 25-27 April, 1983, Boston*. ACM.

[18] Jui-Yi Kao, Narad Rampersad & Jeffrey Shallit (2009): *On NFAs where all states are final, initial, or both*. *Theoret. Comput. Sci.* 410(47-49), pp. 5010–5021. Available at `http://dx.doi.org/10.1016/j.tcs.2009.07.049`.

[19] Ernst L. Leiss (1981): *Succint representation of regular languages by boolean automata*. *Theoret. Comput. Sci.* 13, pp. 323–330. Available at `http://dx.doi.org/10.1016/S0304-3975(81)80005-9`.

[20] A. N. Maslov (1970): *Estimates of the number of states of finite automata*. *Dokl. Akad. Nauk SSSR* 194, pp. 1266–1268 (Russian). English translation: Soviet Math. Dokl. **11** (1970) 1373–1375.

[21] Tomáš Masopust (2010): *Personal communication*.

[22] B. G. Mirkin (1970): *On dual automata*. *Kibernetika (Kiev)* 2, pp. 7–10 (Russian). Available at `http://dx.doi.org/10.1007/BF01072247`. English translation: Cybernetics **2**, (1966) 6–9.

[23] Narad Rampersad (2006): *The state complexity of L^2 and L^k*. *Inform. Process. Lett.* 98(6), pp. 231–234. Available at `http://dx.doi.org/10.1016/j.ipl.2005.06.011`.

[24] Michael Sipser (1997): *Introduction to the theory of computation*. PWS Publishing Company.

[25] Sheng Yu, Qingyu Zhuang & Kai Salomaa (1994): *The state complexities of some basic operations on regular languages*. *Theoret. Comput. Sci.* 125(2), pp. 315–328. Available at `http://dx.doi.org/10.1016/0304-3975(92)00011-F`.